IN THE CLAIMS:

Please amend the claims as follows. No new matter is introduced.

1. (Currently Amended) A method for aligning and inserting data elements into a first memory based upon an instruction sequence consisting of one or more alignment instructions and a single store instruction, comprising the steps of:

given a data item that includes a data element to be stored,

aligning the data element in ~~another~~ a second memory with respect to a predetermined position in the first memory, in response to the one or more alignment instructions;

dynamically generating a mask to enable writing of memory bit lines that correspond to the aligned data element; and

writing the memory bit lines to the first memory under a control of the mask, wherein said generating and writing steps are performed in response to the single store instruction.

2. (Currently Amended) The method of claim 1, wherein the ~~other~~ second memory is a register.

3. (Original) The method of claim 1, further comprising the step of computing the mask from an address argument corresponding to the single store instruction.

4. (Original) The method of claim 3, wherein the address argument comprises a displacement value and an address value.

5. (Original) The method of claim 4, wherein the address value specifies a particular register.

6. (Original) The method of claim 1, further comprising the step of computing the mask based upon a data type of the data element.

7. (Currently Amended) The method of claim 1, wherein the predetermined position in the first memory corresponds to a target position within a memory line.

8. (Original) The method of claim 1, further comprising the step of computing and checking parity information corresponding to the data element.

9. (Original) The method of claim 1, further comprising the step of computing and checking error correction code (ECC) information corresponding to the data element.

10. (Currently Amended) The method of claim 1, further comprising the step of intermediately storing the memory bit lines from the ~~other~~ second memory to a read-write buffer before said writing step.

11. (Original) The method of claim 1, wherein the instruction sequence is without a merge instruction.

12. (Currently Amended) A system for aligning and inserting data elements into a first memory in response to an instruction sequence consisting of one or more alignment instructions and a single store instruction, comprising:

means for receiving a data item that includes a data element to be stored;

means for aligning the data element in ~~another~~ a second memory with respect to a predetermined position in the first memory, in response to the one or more alignment instructions;

means for dynamically generating a mask to enable writing of memory bit lines that correspond to the aligned data element, in response to the single store instruction; and

means for writing the memory bit lines to the first memory under a control of the mask, in response to the single store instruction.

13. (Currently Amended) The system of claim 12, wherein said system exploits partial line write capabilities of the first memory.

14. (Original) The system of claim 12, further comprising logic for computing and checking parity information corresponding to the data element.

15. (Original) The system of claim 12, further comprising logic for computing and checking error correction code (ECC) information corresponding to the data element.

16. (Currently Amended) The system of claim 12, further comprising:

a CPU;

a read-write buffer for intermediately storing, under a control of the CPU, the memory bit lines from the ~~other~~ second memory before said writing step.

17. (Currently Amended) The system of claim 12, wherein the first memory comprises a cache, and said means for writing writes the data element to the cache under the control of the mask.

18. (Original) The system of claim 12, wherein the data item is a data word.

19. (Original) A method for storing data in a memory based upon an instruction sequence consisting of one or more alignment instructions and a single store instruction, comprising the steps of:

aligning the data in a register relative to a location of the data within a target memory address line, in response to the one or more alignment instructions; and

storing a portion of the aligned data within the memory under a control of data type information and an address argument specified by the single store instruction, in response to the single store instruction.

20. (Original) The method of claim 19, wherein said storing step stores the portion of the aligned data under the control of a write mask computed from the data type information and the address specified by the single store instruction.

21. (Original) The method of claim 19, further comprising the step of intermediately storing the aligned data from the register to a read-write buffer before said storing step.

22. (Original) The method of claim 19, wherein the address argument comprises a displacement value and an address value.

23. (Original) The method of claim 22, wherein the address value specifies a particular register.

24. (Original) The method of claim 19, wherein the instruction sequence is without a merge instruction

25-29. (Canceled)